IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with <u>underlining</u> and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (previously amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1 and 24 in accordance with the following:

1.   (currently amended) A method for a computer-aided elimination of at least one inconsistency in a database collection containing a database and at least one copy database of the database, comprising:

changing said database or said at least one copy database, thereby producing an inconsistency;

allocating at least some operations which create an inconsistency to defined conflict types;

allocating each conflict type a decision set which is used to indicate possible decisions which can be used to eliminate an inconsistency created by at least one operation of said respective conflict type;

eliminating said inconsistency utilizing said decision set;

ascertaining a plurality of inconsistencies and their dependencies on one another before eliminating said inconsistency; and

<u>automatically</u> modifying, while eliminating the inconsistency, said decision set for at least one conflict type based on dependencies of said inconsistencies such that a future conflict is eliminated by a modified method.

2.   (previously amended) The method as claimed in claim 1, further comprising the step of eliminating additional inconsistencies.

3.   (previously amended) The method as claimed in claim 1, further comprising the step of allocating each conflict type a decision set which is used to indicate possible decisions which can be used to eliminate an inconsistency created by additional operations of the respective conflict type.

4.   (previously amended) The method as claimed in claim 1, wherein said database

collection contains a plurality of copy databases of said database.

5.   (cancelled)

6.   (previously amended) The method as claimed in claim 1, further comprising the step of ascertaining a conflict, an anomaly, or a pseudo-anomaly when an inconsistency is ascertained.

7.   (cancelled)

8.   (previously amended) The method as claimed in claim 2, further comprising the step of examining, after a prescribable number of eliminated inconsistencies, said database collection for further inconsistencies and their dependencies, anomalies and pseudo-anomalies.

9.   (previously amended) The method as claimed in claim 1, wherein said database collection contains an object-oriented database.

10.   (previously amended) The method as claimed in claim 1, further comprising the step of applying said method in a context of object-oriented software development.

11.   (previously amended) The method as claimed in claim 1, further comprising the step of applying said method in a context of creating a structured electronic document.

12.   (previously amended) A system for eliminating at least one inconsistency in a database collection containing a database and at least one copy database of said database, which inconsistency arises on account of the database or said at least one copy database being changed, comprising:
   a processor configured to:
      allocate at least some operations which create an inconsistency to defined conflict types;
      allocate to each conflict type a decision set which is used to indicate possible decisions which can be used to eliminate an inconsistency created by at least one operation of said respective conflict type; and
      eliminate said inconsistency using said decision set

wherein said processor determines a plurality of inconsistencies and their dependencies on one another before the inconsistency is eliminated;

wherein said processor modifies, during elimination of said inconsistency, a decision set for at least one conflict type based on the dependencies of said inconsistencies such that a future conflict is eliminated by a modified method.

13.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to eliminate a plurality of inconsistencies.

14.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to allocate each conflict type a decision set which is used to indicate possible decisions which can be used to eliminate an inconsistency created by a plurality of operations of said respective conflict type.

15.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to operate a database collection that contains a plurality of copy databases of said database.

16.   (cancelled)

17.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to a certain a conflict, an anomaly or a pseudo-anomaly when an inconsistency is ascertained.

18.   (cancelled)

19.   (previously amended) The system as claimed in claim 13, wherein said processor is configured to examine, after a prescribable number of eliminated inconsistencies, said database collection for further inconsistencies and their dependencies, anomalies and pseudo-anomalies.

20.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to operate on said database collection that contains an object-oriented database.

21.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to operate in a context of object-oriented software development.

22.   (previously amended) The system as claimed in claim 12, wherein said processor is configured to operate in a context of creating a structured electronic document.

23.   (previously amended) A set of a plurality of systems for eliminating at least one inconsistency in a database collection containing a database and at least one copy database of said database, which inconsistency arises on account of said database or said at least one copy database being changed, comprising;

a plurality of processors, wherein each system has at least one processor which is configured to:

allocate at least some operations which create an inconsistency to defined conflict types;

allocate to each conflict type a decision set which is used to indicate possible decisions which can be used to eliminate an inconsistency created by at least one operation of said respective conflict type; and

eliminate said inconsistency using said decision set;

said systems being configured to be coupled to one another to determine a plurality of inconsistencies and their dependencies on one another before the inconsistency is eliminated;

wherein said processors modify, during elimination of said inconsistency, a decision set for at least one conflict type based on the dependencies of said inconsistencies such that a future conflict is eliminated by a modified method.

24. (currently amended) A device comprising:

an identification unit to identify conflicts between a an object-oriented database and a an object-oriented database copy;

. a dependency unit to determine interdependencies between the conflicts;

a solution procedure unit to identify, for each conflict, the type of operation which created the conflict and a solution procedure for the type of operation, to thereby form a set of solution procedures for the respective conflicts;

a modification unit to modify the set of solution procedures based on the interdependencies between the conflicts, to thereby produce a modified set of solution

5

procedures; and

an elimination unit to eliminate the conflicts using the modified set of solution procedures.